DATA PAPER

KB4Rec: A Data Set for Linking Knowledge Bases with Recommender Systems

Wayne Xin Zhao[1†], Gaole He[1], Kunlin Yang[1], Hongjian Dou[1], Jin Huang[1], Siqi Ouyang[2] & Ji-Rong Wen[1]

[1]School of Information, Renmin University of China, Beijing 100872, China

[2]Jacobs Technion-Cornell Institute, Cornell Tech, New York 10044, USA

Keywords: Knowledge-aware recommendation; Recommender system; Knowledge base

Citation: W.X. Zhao, G. He, K. Yang, H. Dou, J. Huang, S. Ouyang & J.-R.Wen. KB4Rec: A data set for linking knowledge bases with recommender systems. Data Intelligence 1(2019), 121-136. doi: 10.1162/dint_a_00008

Received: November 10, 2018; Revised: November 28, 2018; Accepted: December 2, 2018

ABSTRACT

To develop a knowledge-aware recommender system, a key issue is how to obtain rich and structured knowledge base (KB) information for recommender system (RS) items. Existing data sets or methods either use side information from original RSs (containing very few kinds of useful information) or utilize a private KB. In this paper, we present *KB4Rec v1.0*, a data set linking KB information for RSs. It has linked three widely used RS data sets with two popular KBs, namely Freebase and YAGO. Based on our linked data set, we first preform qualitative analysis experiments, and then we discuss the effect of two important factors (i.e., popularity and recency) on whether a RS item can be linked to a KB entity. Finally, we compare several knowledge-aware recommendation algorithms on our linked data set.

1. INTRODUCTION

Recommender systems (RS), which aim to match users with their interested items, have played an important role in various online applications nowadays. Traditional recommendation algorithms mainly focus on learning effective preference models from historical user-item interaction data, e.g., matrix factorization [1]. With the rapid development of Web technologies, various kinds of side information have become available in RSs [2]. At an early stage, the used context information is usually unstructured, and its availability is limited to specific data domains or platforms.

[†] Corresponding author: Wayne Xin Zhao (Email: batmanfly@gmail.com; ORCID: 0000-0002-8333-6196).

More and more efforts have been made recently by both research and industry communities for structuring world knowledge or domain facts in a variety of data domains. One of the most typical organization forms is *knowledge base* (*KB*) [3]. KBs provide a general and unified way to organize and associate information entities, which have been shown to be useful in many applications. For instance, KBs have been used in recommender systems, called *knowledge-aware recommender systems* [4]. To develop a knowledge-aware recommender system, a key issue is how to obtain rich and structured KB information for RS items. Overall, there are two main solutions from existing studies. First, side information has been collected from the RS platform and used as contextual features [5, 6, 7, 8, 9], and some studies further construct tiny and simple KB-like knowledge structure [10, 11, 12]. The number of attributes or relations is usually small, and much useful item information is likely to be missing. Second, several works propose to link RS with private KBs [13, 14, 15]. The linkage results are not publicly available. We are also aware of some closely related studies [16, 17], which aim to link RS items with DBpedia entities. By comparsion, our focus is on Freebase [18] and YAGO [19], which are now widely used in many nature language processing (NLP) or related domains [20, 21, 22].

To address the need for the linked data set of RS and KBs, we present a data set which links two public KBs with recommender systems, named *KB4Rec* v1.0, freely available at https://github.com/RUCDM/KB4Rec. Our basic idea is to heuristically link items from RSs with entities from public large-scale KBs[①]. On the RS side, we select three widely used data sets (i.e., MovieLens [5], LFM-1b [6] and Amazon book [7]) covering three different data domains, namely movie, music and book; on the KB side, we select the two well-known KBs (i.e., Freebase and YAGO). We try to maximize the applicability of our linked data set by selecting very popular RS data sets and KBs. We do not share the original data sets, since they are maintained by original researchers or publishers. These original copies are easily accessible online.

In our KB4Rec v1.0 data set, we have organized the linkage results as linked ID pairs, which consist of a RS item ID and a KB entity ID. All the IDs are inner values from the original data sets. Once such a linkage has been accomplished, it is able to reuse existing large-scale KB data for RSs. For example, the movie “Avatar” from MovieLens data set [5] has a corresponding entity entry in Freebase, and we are able to obtain its attribute information by retrieving all its associated relation triples in Freebase. Based on the linked data set, we first preform some qualitative analysis experiments, and then we discuss the effect of two important factors (i.e., popularity and recency) on whether a RS item can be linked to a KB entity. Finally, we compare several knowledge-aware recommendation algorithms on our linked data set.

With our linkage results and original data copies, it is easy to develop an evaluation set for knowledge-aware recommendation algorithms. We believe such a data set is beneficial to the development of knowledge-aware recommender systems.

[①] We use the terms of “items” and “entities,” respectively, for RSs and KBs.

2. EXISTING DATA SETS AND METHODS

In this section, we briefly review the related data sets and methods.

Early knowledge-aware recommendation algorithms are also called context-aware recommendation algorithms, in which the side information from the original RS platform is considered context data. For example, social network information of Epinions data set is utilized in [23, 24], POI property information of Yelp data set is utilized in [11], movie attribute information of MovieLens data set is utilized in [10] and user profile information of microblogging data set has been utilized in [25, 8]. These data sets usually contain very few kinds of side information, and the relation between different kinds of side information is ignored.

To obtain more structured side information, Heterogeneous Information Networks (HIN) have been proposed as a technique for modeling complex connections between different types of objects [26]. In HINs, we can effectively learn underlying relation patterns (called *meta-path*) and organize side information via meta-path-based representations. For example, HIN-based recommendation systems have been applied to solve PER [10], HeteRecom [27] and MCRec [28]. HIN based algorithms usually rely on graph search algorithms, which is difficult to deal with large-scale relation pattern finding.

More recently, KBs have become a popular kind of data resources to store and organize world knowledge or domain facts. Many studies have been carried out on the construction, inference and applications of KBs [3]. In particular, several pioneering studies [13, 14, 15] try to leverage existing KB information for improving the recommendation performance. They apply a heuristic method for linking RS items with KB entities. In these studies, they use a private KB for linkage, which is not accessible to the public.

We are also aware of some closely related studies [16, 17], which aim to link RS items with KB entities. Nevertheless our focus is on Freebase and YAGO, which are now widely used in many NLP or related domains [20, 21, 22]. Besides, our data sets contain more linked entities and involved relations.

3. LINKED DATA SET CONSTRUCTION

In our work, we need to prepare two kinds of data sets, namely RS and KB. We first describe the original RS and KB data sets and then discuss the linkage method.

3.1 RS Data Sets

Consider three popular RS data sets for linkage, namely *MovieLens*, *LFM-1b* and *Amazon book*, which are from three different domains of movie, music and book, respectively.

- *MovieLens* data set [7] describes users' preferences on movies. A preference record takes the form <user, item, rating, timestamp>, indicating the rating score of a user on a movie at sometime. There have been four MovieLens data sets released, known as 100K, 1M, 10M and 20M, reflecting the approximate number of ratings in each data set. We select the largest MovieLens 20M for linkage.

- *LFM-1b* data set [8] describes users' interaction records on music. It provides information including artists, albums, tracks and users, as well as individual listening events. It records the listening events of a user on songs, but does not contain rating information.
- *Amazon book* data set [9] describes users' preferences on book products, which has a data form, i.e., <user, item, rating, timestamp>. The data set is very sparse, containing 22 million ratings from 8 million users across nearly 23 million items.

The three data sets all provide several kinds of side information such as item titles (all), IMDB ID (movie), writer (book) and artist (music). We utilize such side information for subsequent KB linkage.

3.2 KB Data Sets

We adopt two large-scale pubic KBs, namely *Freebase* and *YAGO*.

Freebase [18] is a KG announced by Metaweb Technologies, Inc. in 2007 and was acquired by Google Inc. on July 16, 2010. Freebase stores facts by triples of the form <head, relation, tail>. Since Freebase shut down its services on August 31, 2016, we use its latest public version.

YAGO [19] is a large semantic KB, which is automatically constructed based on the information of Wikipedia, WordNet, GeoNames and other data sources. It contains 447 million facts about 9.8 million entities in 10 different languages, with an accuracy of above 95% based on manual evaluation. In this paper, we use the version of YAGO in [29].

3.3 RS to KB Linkage

With two KB data sets and three RS data sets, we can form six linkage results. Next, we describe the heuristic method for data linkage.

All three RS data sets provide the information of item titles. For Freebase, with offline KB search APIs, we retrieve KB entities with item titles as queries. Our heuristic linkage method follows the similar idea in [30]. If no KB entity with the exact same title was returned, we say the RS item is *rejected* in the linkage process. If at least one KB entity with the exact same title was returned, we further incorporate one kind of side information as a refined constraint for accurate linkage: *IMBD ID*, *artist name* and *writer name* are used for the three domains of movie, music and book, respectively. We have found only a small number (about 1,000 for each domain) of RS items cannot be accurately linked or rejected via the above procedure, and we simply discard them.

For YAGO, a KB entity is named in a similar way as that in its corresponding Wikipedia URL, in which it is composed of the item title and its related information such as type. For example, film "*Titanic*" is marked as "<*Titanic_*(*1997_film*)>" in YAGO, and the corresponding Wikipedia page can be accessed through the link https://en.wikipedia.org/wiki/Titanic_(1997_film). Therefore, we first compare the title of RS items with the prefix of KB entities. If at least one KB entity was returned, we leverage the "*rdf:type*" relation and suffix (if available) to filter out those entities from other domains. We find that most of the linkage in LFM-1b and Amazon book data sets can be determined accurately (either *linked* or *non-linked*) in this way. By comparison, there exist some ambiguous cases in MovieLens 20M data set, and they are further evaluated through the year restriction.

During the linkage process, we have dealt with several problems that will affect the results of string match algorithms, e.g., lowercase, abbreviation and the order of family/given names. Since the LFM-1b data set is extremely large, we remove all the music items with fewer than 10 listening events. Even after filtering, it still contains about 6.5 million music items.

We present an illustrative example for our linkage results in Figure 1. In this example, there are two pairs of an item from MovieLens 20M and its linked entity from Freebase. The two movie items are "*Spider man*" and "*Spider man 2*." It is clear to see that both movies share many common attributes in Freebase. With such linkage results, it is easy to obtain rich KB information about RS items, which are likely to be useful in recommendation performance.

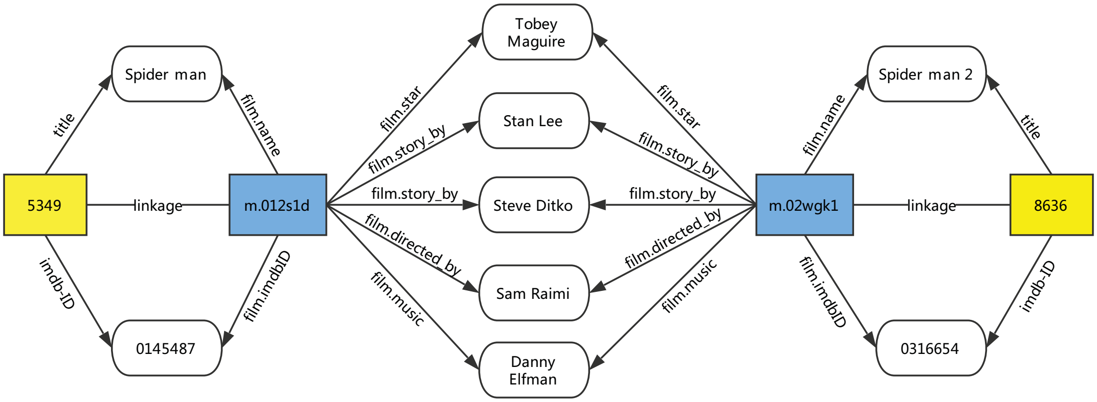


Figure 1. Linkage example of MovieLens 20M items with Freebase entities. Note: We highlight the MovieLens IDs and Freebase IDs.

3.4 Basic Statistics

We summarize the basic statistics of the three linked data sets in Table 1. It can be observed that for the MovieLens 20M data set, we have a very high linkage ratio: about 95.2% or 79.5% items can be accurately linked to an entity from Freebase or YAGO. But for the rest two domains, the linkage ratios are very low, especially using YAGO for linkage. MovieLens 20M data set has a high linkage ratio, which is probably because that it contains fewer items than the other two data sets, which themselves are refined by original releasers. Besides, we speculate that there may be some domain bias in the construction of KBs. Overall, more RS items can be linked with Freebase entities than YAGO. Although the linkage ratios for the latter two data sets are not high, the absolute numbers of linked items are large. We also report the number of overlapping *linked entities* for the two KBs in the last row of Table 1. We can see that there are also more linked items in the movie domain. Such a linked data set is feasible for research-purpose studies.

Table 1. Statistics of the linkage results.

Data sets	Numbers	MovieLens 20M	LFM-1b	Amazon book
RS data sets	#users	138,493	120,317	3,468,412
	#items	27,279	6,479,700	2,330,066
	#interactions	20,000,263	1,021,931,544	22,507,155
Freebase	#linked-items	25,982	1,254,923	109,671
	Linkage ratio	95.2%	19.4%	4.7%
YAGO	#linked-items	21,688	49,608	17,607
	Linkage ratio	79.5%	0.8%	0.8%
Overlap	#overlap	21,221	26,126	7,398

Note: The three domains correspond to the RS data sets of MovieLens 20M, LFM-1b and Amazon book, respectively.

3.5 Shared Data Sets

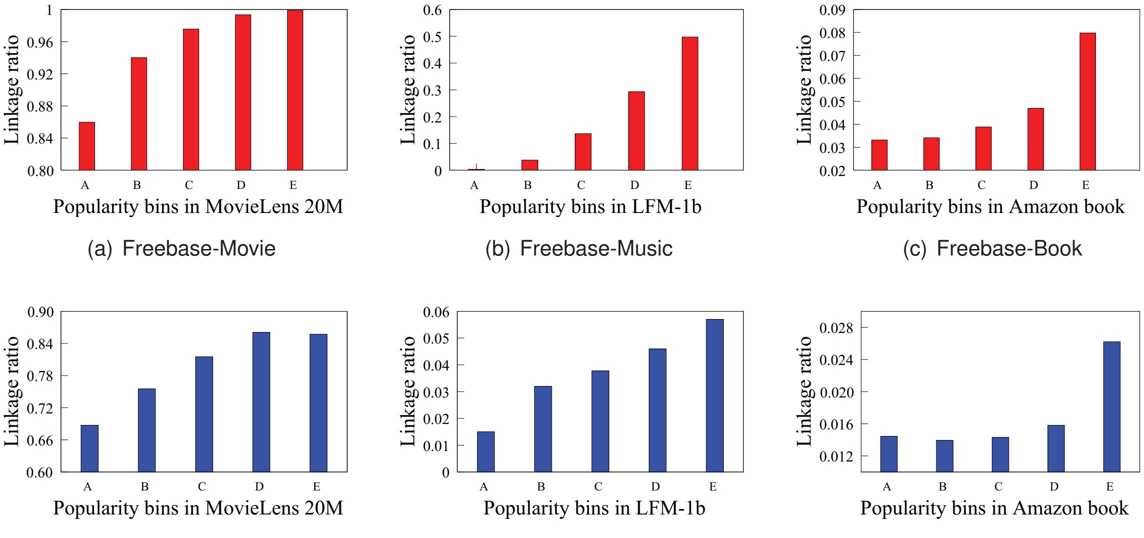
We name the above linked KB data set for recommender systems as *KB4Rec* v1.0, freely available at https://github.com/RUCDM/KB4Rec. In our KB4Rec v1.0 data set, we organized the linkage results by linked ID pairs, which consist of a RS item ID and a KB entity ID. All the IDs are inner values from the original data sets. For Freebase, we have 25,982, 1,254,923 and 109,671 linked ID pairs for MovieLens 20M, LFM-1b and Amazon book, respectively; for YAGO, we have 21,688, 49,608 and 17,607 linked ID pairs for MovieLens 20M, LFM-1b and Amazon book, respectively.

4. LINKAGE ANALYSIS

Previously, we have shown the linkage ratios for different data sets. We find that a considerable amount of RS items cannot be linked to KB entities. It is interesting to study what factors will affect the linkage ratio. We consider two factors for analysis.

4.1 Effect of Popularity on Linkage

Intuitively, a popular RS item should be more likely to be included in a KB than an unpopular item, since it is reasonable to incorporate more “important” RS items rated by the RS users into KBs. The construction of KB itself usually involves manual efforts, which is difficult to avoid the bias of human attention. To measure the popularity of a RS item, we adopt a simple frequency-based method by counting the number of users who have interacted with the item. This measure characterizes the attractiveness of an item from the users in a RS. First, we sort the items ascendingly according to its popularity value. Then, we further equally divide all the items into five ordered bins with the same number in each bin. Hence, an item with a larger bin number will be more popular than another with a smaller bin number. Then, we compute the linkage ratio for each bin and the results are reported in Figure 2. It can be observed that a bin with a larger number has a higher linkage ratio than the ones with a smaller number. The results indicate that popularity is likely to have a positive effect on linkage.



(a) Freebase-Movie

(b) Freebase-Music

(c) Freebase-Book

(d) YAGO-Movie

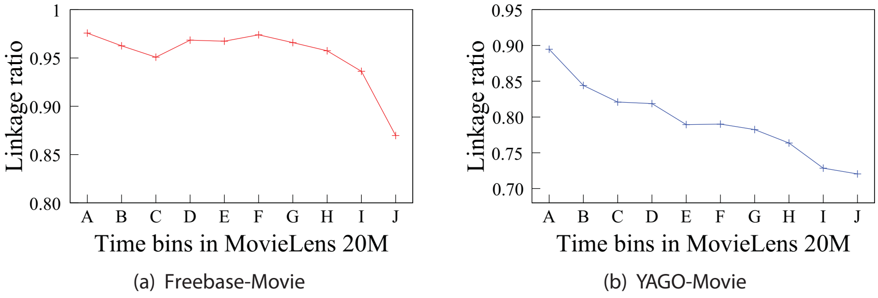
(e) YAGO-Music

(f) YAGO-Book

Figure 2. Examining the effect of popularity on the linkage results. Note: We use A, B, … to indicate the bin number in an ordered way. The first three subfigures correspond to the popularity analysis on Freebase, and the last three subfigures correspond to the popularity analysis on YAGO.

4.2 Effect of Recency on Linkage

The second factor we consider is the recency, i.e., the time when a RS item was created. Our assumption is that if a RS item was created or released on an earlier time, it would be more probable to be included in KBs. Since human attention aggregation is a gradually growing process, a RS item usually requires a considerable amount of time to become popular. To check this assumption, we need to obtain the release date of RS items. However, only the MovieLens 20M data set contains such an attribute information, so we only report the analysis result on this data set. We first sort the items according to their release dates ascendingly, and then equally divide all the items into 10 ordered bins following the procedure of the above popularity analysis. Finally, we compute the linkage ratios for each bin. The results are reported in Figure 3. We can see that the linkage ratios gradually decrease with time going by. The results indicate that recency is likely to have a negative effect on linkage, i.e., an older RS item seems to be more probable to be included in a KB than a more recent one. In Figure 3 (a), the last bin has a dramatic drop, since our version of MovieLens is April 2015.



(a) Freebase-Movie (b) YAGO-Movie

Figure 3. Examining the effect of recency on the linkage results. Note: We use A, B, … to indicate the bin number in an ordered way. The first subfigure corresponds to the recency analysis on Freebase, and the second subfigure corresponds to the recency analysis on YAGO.

The above analysis has indicated that both popularity and recency have a considerable effect on the final linkage results. However, the construction process of KB is very complicated, and many important factors will affect this process. For future research, it is worth delving into what are other important factors and how they affact the construction process of KB.

5. EXPERIMENT

In this section, we present the comparison of some existing recommendation algorithms using our linked data sets.

5.1 Experimental Setup

Our purpose is to test whether the incorporated KB information is useful to improve the recommendation performance. In Freebase, there are more linked entities and associated relations. So we only adopt the linked data set of Freebase for evaluation, and the results from YAGO are similar and omitted here.

The original linked data set is very large, so we first generate a small evaluation set for the following experiments. We took the subset from the last year for LFM-1b data set and the subset from year 2005 to 2015 for MovieLens 20M data set. We also perform 3-core filtering for Amazon book data set and 10-core filtering for other data sets. This part mainly follows the preprocessing step in [31]. And then, we have kept items which are linked by our data set. We report the statistics of data sets in Table 2.

Table 2. Statistics of the evaluation data sets for the Freebase KB.

Data sets	#users	#items	#interactions
MovieLens 20M	61,583	19,533	5,868,015
LFM-1b	7,694	30,658	203,975
Amazon book	65,125	69,975	828,560

Note: In this data set, all the items are linked with Freebase.

Following [32], we consider the last-item recommendation task for evaluation. We set up such a task since it is a commonly used evaluation setting for RSs, and it is easy to compare different methods. Given a user, first we sort the items according to the interaction timestamp ascendingly, and then we take the last item into the test set and the rest into training set. The final goal is to predict the last item given the previous interaction sequence of a user. Since enumerating all the items as candidate is time-consuming, we pair each ground-truth with 100 negative items to form a randomly ordered list. Then each comparison method is to return a ranked list according to its recommendation confidence. To evaluate different methods, we adopt a variety of evaluation metrics, including the Mean Reciprocal Rank (MRR), Hit Ratio (HR) and Normalized Discounted cumulative gain (NDCG).

5.2 KB Information Representation

Our focus is to provide rich KB information for recommender systems. A simple way is to represent KB information with a one-hot vector, which is sparse and large. Here we borrow the idea in [15, 33] to embed KB data into low-dimensional vectors. Then the learned embeddings are used for subsequent recommendation algorithms. To train TransE [33], we start with linked entities as seeds and expand the graph with one-step search. As not all the relations in KBs are useful, we remove unfrequent and general-purpose relations together with all their associated KB triples. After that, each linked item is associated with a learned KB embedding vector. We report the statistics for training TransE in Table 3.

Table 3. Statistics of our subgraph for training TransE.

Data sets	#entities	#relations
MovieLens 20M	1,125,099	81
LFM-1b	214,524	19
Amazon book	313,956	49

Note: #entities indicates the number of entities that are extended by seed entities with one-step search in Freebase.

5.3 Methods to Compare

We consider the following methods for performance comparison[2]:

- **BPR** [34]: It learns a matrix factorization model by minimizing the pairwise ranking loss in a Bayesian framework.
- **SVDFeature** [35]: It is a model for feature-based collaborative filtering. In this paper we use the KB embeddings as context features to feed into SVDFeature.

[2] Here, since our purpose is to illustrate the use of this linked data set, we only select four methods for performance comparison. We will try more knowledge-ware recommendation algorithms in our future work.

- **mCKE** [13]: It first proposes to incorporate KB and other information to improve the recommendation performance. For fairness, we implement a simplified version of CKE by only using KB information, and exclude image and text information. Different from the original CKE, we fix KB representations and adopt the learned embeddings by TransE.
- **KSR** [31]: It is a Knowledge-enhanced Sequential Recommender (KSR). It incorporates KB information to enhance the semantic representation memory networks.

5.4 Results and Analysis

The results of different methods for the last-item recommendation are presented in Table 4. We can see that:

1). Among all the methods, BPR performs worst on the first two data sets, but very well on the Amazon book data set. A possible reason is the first two data sets are relatively dense while the Amazon book data set is sparse. A lightweight method is likely to obtain a better performance than more complicated methods on a sparse data set.

2). SVDFeature is implemented with a pairwise ranking loss function, and it can be roughly understood as an enhanced BPR model with the incorporation of the learned KB embeddings. Compared with BPR, SVDFeature is slightly better on the MovieLens 20M data set, substantially better on the LFM-1b data set, but worse on the Amazon book data set. In SVDFeature, each context feature will incorporate some number of parameters (deciding on the number of dimensions). Hence, on a sparse data set, it may not work better than the simple BPR model.

3). Next, we analyze the performance of the knowledge-aware recommendation methods, namely mCKE and KSR. Overall, mCKE does not work well as expected, which only has a good performance on the LFM-1b data set. A possible reason is that our implementation of mCKE fixes the learned KB embeddings, while the original CKE model adaptively updates KB embeddings. As a comparison, the recently proposed KSR method works best consistently on the three data sets. KSR combines the capacity of modeling data sequences from Recurrent Neural Networks (RNN) and the capacity of storing data in a long term from Memory Networks (MN). It further enhances MNs with the learned KB embeddings.

Table 4. Performance comparison of different methods on the task of last-item recommendation.

Data sets	Methods	MRR	Hit@10	NDCG@10
MovieLens 20M	BPR	0.128	0.276	0.144
	SVDFeature	0.204	0.448	0.243
	mCKE	0.178	0.382	0.209
	KSR	0.294	0.571	0.344
LFM-1b	BPR	0.227	0.458	0.265
	SVDFeature	0.337	0.544	0.373
	mCKE	0.371	0.541	0.399
	KSR	0.427	0.607	0.460
Amazon book	BPR	0.222	0.505	0.272
	SVDFeature	0.264	0.544	0.315
	mCKE	0.248	0.494	0.291
	KSR	0.353	0.653	0.413

6. CONCLUSION

In this paper, we present *KB4Rec v1.0*, a data set linking KB information for recommender systems. It has linked three widely used RS data sets with the popular KBs Freebase [18] and YAGO [19]. Based on our linked data set, we first preform some qualitative analysis experiments, and then we discuss the effect of two important factors (i.e., popularity and recency) on whether a RS item can be linked to a KB entity. Finally, we compare several knowledge-aware recommendation algorithms on our linked data set.

For future work, we will consider linking more RS data sets with KBs. We will also test the performance of more knowledge-aware recommendation algorithms on more recommendation tasks using the linked data set.

AUTHOR CONTRIBUTIONS

W.X. Zhao (batmanfly@gmail.com) and J.-R. Wen (jrwen@ruc.edu.cn) led the whole work. W.X. Zhao organized the content and wrote the paper. G. He (hegaole@ruc.edu.cn), H. Dou (hongjiandou@ruc.edu.cn) and J. Huang (jin.huang@ruc.edu.cn) generated the linkage results for Freebase and run the experimental results; K. Yang (kunliny@ruc.edu.cn) generated the linkage results for YAGO. All the authors have made meaningful and valuable contributions in revising and proofreading the manuscript.

ACKNOWLEDGEMENTS

The work was partially supported by National Natural Science Foundation of China (No. 61872369, No. 61832017 and No. 61502502).

REFERENCES

[1] Y. Koren, R. Bell, & C. Volinsky. Matrix factorization techniques for recommender systems. Computer 42(8) (2009), 30–37. doi: 10.1109/MC.2009.263.

[2] S. Rendle. Factorization machines with libFM. ACM Transactions on Intelligent Systems and Technology 3(3) (2012), Article No. 57. doi: 10.1145/2168752.2168771.

[3] Q. Wang, Z. Mao, B.Wang, & L. Guo. Knowledge graph embedding: A survey of approaches and applications. IEEE Transactions on Knowledge and Data Engineering 29(12)(2017), 2724–2743. doi: 10.1109/TKDE.2017.2754499.

[4] S. Bouraga, I. Jureta, S. Faulkner, & C. Herssens. Knowledge-based recommendation systems: A survey. International Journal of Intelligent Information Technologies 10(2)(2014), 1–19. doi: 10.4018/ijiit.2014040101.

[5] F. M. Harper, & J. A. Konstan. The MovieLens data sets: History and context. ACM Transactions on Interactive Intelligent Systems 5(4)(2016), Article No. 19. doi: 10.1145/2827872.

[6] M. Schedl. The lfm-1b data set for music retrieval and recommendation. In: Proceedings of the 2016 ACM International Conference on Multimedia Retrieval, 2016, pp. 103–110. doi: 10.1145/2911996.2912004.

[7] R. He, & J. Mcauley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In: Proceedings of the 25th International Conference on World Wide Web, 2016, pp. 507–517. doi: 10.1145/2872427.2883037.

[8] W.X. Zhao, S. Li, Y. He, E.Y. Chang, J.-R. Wen, & X. Li. Connecting social media to e-commerce: Cold-start product recommendation using microblogging information. IEEE Transactions on Knowledge and Data Engineering 28(5)(2016), 1147–1159. doi: 10.1109/TKDE.2015.2508816.

[9] J. Huang, Z. Ren, W.X. Zhao, G. He, J. Wen, & D. Dong. Taxonomy-aware multi-hop reasoning networks for sequential recommendation. In: Proceedings of the 12th ACM International Conference on Web Search and Data Mining (WSDM'19), 2019, pp. 573–581. doi: 10.1145/3289600.3290972.

[10] X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, & J. Han. Personalized entity recommendation: A heterogeneous information network approach. In: Proceedings of the 7th ACM International Conference on Web Search and Data Mining, 2014, pp. 283–292. doi: 10.1145/2556195.2556259.

[11] H. Gao, J. Tang, X. Hu, & H. Liu. Content-aware point of interest recommendation on location-based social networks. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence, 2015, pp. 1721–1727. Available at: https://dl.acm.org/citation.cfm?id=2886559.

[12] C. Shi, B. Hu, W.X. Zhao, & P.S. Yu. Heterogeneous information network embedding for recommendation. IEEE Transactions on Knowledge and Data Engineering 31(2)(2019), 357–370. doi: 10.1109/TKDE.2018.2833443.

[13] F. Zhang, N.J. Yuan, D. Lian, X. Xie, & W. Ma. Collaborative knowledge base embedding for recommender systems. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 353–362. doi: 10.1145/2939672.2939673.

[14] H. Wang, F. Zhang, X. Xie, & M. Guo. DKN: Deep knowledge-aware network for news recommendation. In: The 2018 World Wide Web Conference, 2018, pp. 1835–1844. doi: 10.1145/3178876.3186175.

[15] H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie, & M. Guo. Ripple network: Propagating user preferences on the knowledge graph for recommender systems. arXiv preprint. arXiv: 1803.03467v4.

[16] T.D. Noia, V.C. Ostuni, P. Tomeo, & E.D. Sciascio. Sprank: Semantic path-based ranking for top-n recommendations using linked open data. ACM Transactions on Intelligent Systems and Technology 8(1)(2016), Article No. 9. doi: 10.1145/2899005.

[17] T.D. Noia, & V.C. Ostuni. Recommender systems and linked open data. In: W. Faber, & A. Paschke (eds.) Reasoning Web 2015: Reasoning Web. Web Logic Rules. Cham, Switzerland: Springer International Publishing, 2015, pp. 88–113. doi: 10.1007/978-3-319-21768-0_4.

[18] Google. Freebase data dumps. Available at: https://developers.google.com/freebase/data.

[19] F.M. Suchanek, G. Kasneci, & G. Weikum. YAGO: A core of semantic knowledge. In: The 16th International Conference on the World Wide Web, 2007, pp. 697–706. doi: 10.1145/1242572.1242667.

[20] H. Bast, & E. Haussmann. More accurate question answering on Freebase. In: Proceedings of the 24th ACM International Conference on Information and Knowledge Management, 2015, pp. 1431–1440. doi: 10.1145/2806416.2806472.

[21] W. Cui, Y. Xiao, & W. Wang. KBQA: An online template based question answering system over Freebase. In: Proceedings of the 25th International Joint Conference on Artificial Intelligence, 2016, pp. 4240–4241. Available at: https://dl.acm.org/citation.cfm?id=3061256.

[22] P. Adolphs, M. Theobald, U. Schafer, H. Uszkoreit, & G. Weikum. YAGO-QA: Answering questions by structured knowledge queries. In: Proceedings of the 2011 IEEE International Conference on Semantic Computing, 2011, pp. 158–161. doi: 10.1109/ICSC.2011.30.

[23] M. Jamali, & M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In: Proceedings of the 4th ACM Conference on Recommender Systems, 2010, pp. 135–142. doi: 10.1145/1864708.1864736.

[24] H. Ma, I. King, & M.R. Lyu. Learning to recommend with social trust ensemble. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2009, pp. 203–210. doi: 10.1145/1571941.1571978.

[25] W.X. Zhao, Y. Guo, Y. He, H. Jiang, Y. Wu, & X. Li. We know what you want to buy: A demographic-based system for product recommendation on microblogs. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 1935–1944. doi: 10.1145/2623330.2623351.

[26] Y. Sun, & J. Han. Mining heterogeneous information networks: A structural analysis approach. ACM SIGKDD Explorations Newsletter 13(2)(2012), 20–28. doi: 10.1145/2481244.2481248.

[27] C. Shi, C. Zhou, X. Kong, P.S. Yu, G. Liu, & B. Wang. Heterecom: A semantic-based recommendation system in heterogeneous networks. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2012, pp. 1552–1555. doi: 10.1145/2339530.2339778.

[28] B. Hu, C. Shi, W.X. Zhao, & P.S. Yu. Leveraging meta-path based context for top-N recommendation with a neural co-attention model. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018, pp. 1531–1540. doi: 10.1145/3219819.3219965.

[29] J. Hoffart, F.M. Suchanek, K. Berberich, & G. Weikum. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. Artificial Intelligence 194 (2013), 28–61. doi: 10.1016/j.artint.2012.06.001.

[30] T. Scheffler, R. Schirru, & P. Lehmann. Matching points of interest from different social networking sites. In: Proceedings of the 35th Annual German Conference on Advances in Artificial Intelligence, 2012, pp. 245–248. doi: 10.1007/978-3-642-33347-7_24.

[31] J. Huang, W.X. Zhao, H. Dou, J. Wen, & E.Y. Chang. Improving sequential recommendation with knowledge-enhanced memory networks. In: The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'18), 2018, pp. 505–514.

[32] R. He, W. Kang, & J. McAuley. Translation-based recommendation: A scalable method for modeling sequential behavior. In: Proceedings of the 11th ACM Conference on Recommender Systems, 2017, pp. 161–169. doi: 10.1145/3109859.3109882.

[33] A. Bordes, N. Usunier, A. García-Durán, J. Weston, & O. Yakhnenko. Translating embeddings for modeling multi-relational data. In: The 27th Conference on Neural Information Processing Systems (NIPS 2013), 2013, pp. 2787–2795. Available at: http://papers.nips.cc/paper/5071-translating-embeddings-for-modeling-multi-relational-data.pdf.

[34] S. Rendle, C. Freudenthaler, Z. Gantner, & L. Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In: Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, 2009, pp. 452–461. Available at: https://dl.acm.org/citation.cfm?id=1795114.1795167

[35] T. Chen,W. Zhang, Q. Lu, K. Chen, Z. Zheng, & Y. Yu. Svdfeature: A toolkit for feature-based collaborative filtering. The Journal of Machine Learning Research 13(1)(2012), 3619–3622.

AUTHOR BIOGRAPHY

Wayne Xin Zhao is currently an Associate Professor at School of Information, Renmin University of China. He received his PhD degree from Peking University in 2014. His research interests are recommender systems and natural language processing. He has published more than 50 referred papers in international conferences and journals.

Gaole He is currently a graduate student at School of Information, Renmin University of China. He received his Bachelor's degree from School of Information, Renmin University of China in 2018. His research mainly focuses on knowledge graph, deep learning and network embedding.

Kunlin Yang is currently an undergrduate student at School of Information, Renmin University of China. He is expected to receive his Bachelor's degree in Engineering in 2019. His research interests include recommender systems and knowledge graph.

Hongjian Dou is currently a graduate student at School of Information, Renmin University of China. He is working at the Key Laboratory of Big Data Management and Analysis Methods, Beijing. His research mainly focuses on natural language processing, deep learning and recommender systems.

Jin Huang is currently a graduate student at School of Information, Renmin University of China. She is working at the Key Laboratory of Big Data Management and Analysis Methods, Beijing. Her research interests include recommender systems, deep learning and knowledge base.

Siqi Ouyang is currently a graduate student at Jacobs Technion-Cornell Institute, Cornell Tech. She received her Bachelor's degree from Renmin University of China in 2018. Her research mainly focuses on deep learning in natural language processing and recommendation systems.

Ji-Rong Wen is a Professor at School of Information, Renmin University of China. He is also the Director of the Key Laboratory of Big Data Management and Analysis Methods, Beijing. Before that, he was a senior researcher and group manager of the Web Search and Mining Group at Microsoft Research Asia (MSRA). His main research interests include big data management and analytics, information retrieval, data mining and machine learning. He is currently the associate editor of *ACM Transactions on Information Systems* (TOIS). He is a senior member of the IEEE.